**Burrows-Wheeler Transform (BWT):**

googol$

1234567

Order = $, g, l, o

Start by creating all transformations by shifting end letter to the start

googol$, $googol, l$googo, ol$goog, gol$goo, ogol$go, oogol$g

Sort them based on lexigraphical ordering:

$googol > gol$goo > googol$ > l$googo > ogol$go > ol$goog > oogol$g

The last column of letter (lo$oogg) is the BWT(googol$)

Need to index the occurrence of the letters in the BWT:

IE: (l, 0), (o, 0), ($, 0), (o, 1), (o, 2), (g, 0), (g, 1)

Need to note the number of character of lower lexical order for each letter in the BWT:

IE: C[i] = ($, 0), (g, 1), (l, 3), (o, 4)

**Inverting the BWT:**

Given the BWT, indices, and number of characters of lower lex order:

Start with (l, 0):

Look up C[l] + 0 = 3

Now go to the row that corresponds to the index of 3

IE: (o, 1)

Start with (o, 1):

Look up C[o] + 1 = 5

Start with (g, 0):

Look up C[g] + 0 = 2

**Example (purple):**

Order = $, e, l, p, r, u

purple$, $purple, e$purpl, le$purp, ple$pur, rple$pu, urple$p

Sort:

$purple, e$purpl, le$purp, ple$pur, purple$, rple$pu, urple$p = BWT = elpr$up

**Example:**

NVSE$E

$, e, n, s, v

C = ($, 0), (e, 1), (n, 3), (s, 4), (v, 5)

(N, 0) = 3+0 = 3

(e, 0) = 1 + 0 = 1

(v, 0) = 5 + 0 = 5

(e, 1) = 1 + 1 = 2

(s, 0) = 4 + 0 = 4

($, 0) = 0 + 0 = 0

Result = $seven = seven

**Search via BWT:**

Say we wanted to search for “go”